REMARKS

The Examiner has rejected Claims 1-5, 7-14, 16-23 and 25-27 under 35 U.S.C. 102(e) as being anticipated by Munson et al. (U.S. Patent No. 6,681,331). Applicant respectfully disagrees with such rejection, especially in view of the amendments made hereinabove to each of the independent claims. Specifically, applicant has amended each of the independent claims to incorporate the subject matter of Claim 2 et al.

With respect to each of the independent claims, the Examiner has relied on the following excerpts to make a prior art showing of applicant's claimed "producing a run-time execution profile by gathering statistics related to execution of a protocol stack within the kernel of the server; wherein the protocol stack processes packets received from client computer systems; comparing the run-time execution profile with a normal execution profile for the kernel of the server; wherein the normal execution profile is representative of execution when the server is not subject to a denial-of-service attack; and indicating that a denial-of-service attack is taking place if the run-time execution profile deviates from the normal execution profile" (see the same or similar, but not identical language in each of the independent claims).

> "Also, this prior art system does not allow monitoring of all types of software activity, since it is limited to operating system kernel events. Accordingly, it would be desirable to provide a real time intrusion detection paradigm that is applicable to monitoring almost any type of program." (Col. 1, lines 54-57)
>
> "Each program module, M.sub.i, of a plurality of program modules 101 will have calls placed in it at each entry point and before each return. Control is passed to a mapping module 102 that records any transition into and out of a program module. The mapping module transmits the module transitions to a module sequence buffer 103 that buffers these data until they are requested from the external program environment. All of structures 101-103 are encapsulated within the operating environment of a program to which the present invention is applied to detect anomalous behavior or an intrusion.
>
> FIG. 2 shows the operation of a first profile transducer 202. It is the purpose of first profile transducer 202 to capture module sequence information 201 from the internally instrumented program environment. At intervals determined by an externally provided

> sampling engine 204, first profile transducer 201 interrogates module sequence buffer 103, requesting current profile information. The profile information obtained from the module sequence buffer is a list of all modules that have executed since the last interrogation, and the frequencies of their executions. First profile transducer 202 normalizes each of the module frequencies by dividing them by the total number of module transitions that have occurred during the sampling interval. These execution profiles are transmitted to and retained by an execution profile buffer 203." (Col. 3, lines 34-59-emphasis added)
>
> "FIG. 5 shows the operation of an execution profile comparator 502. The execution profile comparator determines any difference (i.e., a differenced profile) between a current execution profile 501 most recently obtained from first profile transducer 202 and a nominal execution profile obtained from nominal profiles data 506, which represents the steady-state behavior of the software system with no intrusive activity. The nominal profiles data are initially established by a calibration process that is implemented by running the program in a calibration mode in which the program is run through as many of the functions and operations performed during a nominal operational phase." (Col. 4, lines 26-37)

First, applicant respectfully asserts that the above excerpts from Munson completely fail to teach applicant's claimed "producing a run-time execution profile by gathering statistics related to execution of a protocol stack within the kernel of the server" (emphasis added). After reviewing such excerpts, applicant notes that Munson only teaches a "nominal execution profile...which represents the steady-state behavior of the software system with no intrusive activity." Clearly, a profile of the steady-state of the software system, as disclosed in Munson, does not meet a profile produced by "gathering statistics related to execution of a protocol stack within the kernel of the server," as claimed by applicant (emphasis added).

Second, applicant respectfully asserts that the above excerpts from Munson fail to teach applicant's claimed technique "wherein the protocol stack processes packets received from client computer systems." Applicant notes that simply nowhere in Munson is there any disclosure of a protocol stack that processes packets. In fact, Munson only teaches program modules that have calls placed on them along with a module sequence buffer that simply buffers module transitions until they are requested (see emphasized excerpt above), but does not specifically teach a protocol stack that processes packets.

Third, applicant respectfully asserts that the above excerpts from Munson also fail to teach applicant's claimed "comparing the run-time execution profile with a normal execution profile for the kernel of the server." Again, applicant emphasizes that Munson teaches determining the difference "between a current execution profile 501...and a nominal execution profile" where the nominal execution profile only represents the steady state behavior of the software system, and therefore does not teach a specific "profile for the kernel of the server," as claimed by applicant (emphasis added).

The Examiner is reminded that a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. Of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). Moreover, the identical invention must be shown in as complete detail as contained in the claim. *Richardson v. Suzuki Motor Co.*868 F.2d 1226, 1236, 9USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim.

This criterion has simply not been met by the Munson reference, as noted above. Nevertheless, despite such paramount deficiencies and in the spirit of expediting the prosecution of the present application, applicant has substantially incorporated the subject matter of Claim 2 et al. into each of the independent claims.

With respect to the subject matter of Claim 2 et al., the Examiner has relied on Col. 3, lines 34-59 and Col. 6, lines 14-25 in Munson to make a prior art showing of applicant's claimed technique "wherein producing the run-time execution profile involves gathering statistics regarding a fraction of time that the server spends executing one or more portions of code related to the protocol stack." Applicant respectfully asserts that the intervals disclosed by Munson simply relate to intervals for requesting current profile information by a first profile transducer and intervals for obtaining data by a comparator. Clearly, such intervals do not meet a "fraction of time that the server spends

executing one or more portions of code related to the protocol stack," as claimed by applicant (emphasis added).

Again, since the Munson reference fails to teach all of applicant's claim limitations, especially in view of the amendments made hereinabove to each of the independent claims, a notice of allowance or a specific prior art showing of each of the foregoing claimed features, in combination with the remaining claimed features, is respectfully requested.

Applicant further notes that the prior art is also deficient with respect to the dependent claims. Just by way of example, with respect to Claim 3 et al., the Examiner has relied on Col. 4, lines 26-65 in Munson to make a prior art showing of applicant's claimed technique "wherein producing the run-time execution profile involves producing a vector indicating a number of times that the server is found to be executing the one or more portions of code related to the protocol stack."

Applicant respectfully asserts that such excerpt merely relates to a nominal execution profile that "represents steady-state behavior...with no intrusive activity" and not a run-time execution profile, in the context claimed by applicant. Furthermore, such excerpt discloses utilizing the difference between a current execution profile and the nominal execution profile. Clearly, such teachings do not even suggest producing a run-time execution profile, let alone any sort of "vector indicating a number of times that the server is found to be executing the one or more portions of code related to the protocol stack," as claimed by applicant

With respect to Claim 4 et al., the Examiner has relied on Col. 3, lines 34-59 in Munson to make a prior art showing of applicant's claimed technique "wherein the one or more portions of code related to the protocol stack include: a portion related to processing TCP SYN requests; a portion related to processing TCP ACKs; a portion related to processing TCP data; a portion related to processing ICMP echo requests; and a portion that is unrelated to the protocol stack."

Applicant respectfully asserts that such excerpt only teaches a module sequence buffer that buffers data until it is requested, and not a protocol stack in the context claimed by applicant. Furthermore, Munson only generally teaches calls made to each program module, but not specifically a server that executes portions of code including "a portion related to processing TCP SYN requests; a portion related to processing TCP ACKs; a portion related to processing TCP data; a portion related to processing ICMP echo requests; and a portion that is unrelated to the protocol stack," as claimed.

With respect to Claim 8 et al., the Examiner has relied on Col. 3, lines 34-59 and Col. 6, lines 14-25 in Munson to make a prior art showing of applicant's claimed "gathering statistics for a concurrent execution profile over a concurrent time window that overlaps the first time window and the second time window, so that a denial-of service attack that overlaps the first time window and the second time window can be detected in the concurrent time window." Applicant respectfully asserts that such excerpts only generally teach intervals at which data is requested, but not specifically that such intervals are "over a concurrent time window that overlaps the first time window and the second time window, so that a denial-of service attack that overlaps the first time window and the second time window can be detected in the concurrent time window," as claimed by applicant (emphasis added).

Again, since the Munson reference fails to teach all of applicant's claim limitations, as noted above, a notice of allowance or a specific prior art showing of each of the foregoing claimed features, in combination with the remaining claimed features, is respectfully requested.

Still yet, applicant brings to the Examiner's attention the subject matter of new Claims 28-29 below, which are added for full consideration:

"wherein the protocol stack includes a datalink layer, an Internet Protocol layer, a Transmission Control Protocol/User Datagram Protocol/Internet Control

Message Protocol layer and an application layer such that the one or more portions of code are associated with each of the layers" (see Claim 28); and

"wherein producing the normal execution profile involves producing a vector indicating a normal number of times that the server is found to be executing the one or more portions of code related to the protocol stack" (see Claim 29).

Thus, all of the independent claims are deemed allowable. Moreover, the remaining dependent claims are further deemed allowable, in view of their dependence on such independent claims.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 505-5100. The Commissioner is authorized to charge any additional fees or credit any overpayment to Deposit Account No. 50-1351 (Order No. NAI1P261/01.170.01).

Respectfully submitted,
Zilka-Kotab, PC.

Kevin J. Zilka
Registration No. 41,429

P.O. Box 721120
San Jose, CA 95172-1120
408-505-5100